(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2001 (13.09.2001)

PCT

(10) International Publication Number
WO 01/67218 A1

(51) International Patent Classification[7]: G06F 1/24

(21) International Application Number: PCT/US01/07381

(22) International Filing Date: 8 March 2001 (08.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/520,404 8 March 2000 (08.03.2000) US

(71) Applicant *(for all designated States except US)*: **SHUFFLE MASTER, INC.** [US/US]; 10901 Valley View Road, Eden Prairie, MN 55344 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **JACKSON, Mark, D.** [US/US]; 907 Canosa Court, Fort Collins, CO 80526 (US). **MARTINEK, Michael, G.** [US/US]; 3007 Indigo Circle North, Fort Collins, CO 80528 (US).

(74) Agent: **DICKE, Steven, E.**; Dicke, Billig & Czaja, P.A., Suite 1250, 701 Fourth Avenue South, Minneapolis, MN 55415 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/67218 A1

(54) Title: ENCRYPTION IN A SECURE COMPUTERIZED GAMING SYSTEM

(57) Abstract: The present invention provides an architecture and method for a gaming-specific platform that features secure storage (354) and verification (366) of game code and other data, provides the ability to securely exchange data with a computerized wagering gaming system, and does so in a manner that is straightforward and easy to manage. Some embodiments of the invention provide the ability to identify game program code as certified or approved, such as by the Nevada Gaming Regulations Commission or other regulatory agency. The invention provides these and other functions by use of encryption (216), including digital signatures (220) and hash functions (210) as well as other encryption methods.

ENCRYPTION IN A SECURE COMPUTERIZED GAMING SYSTEM

Field of the Invention

The invention relates generally to computerized wagering game systems, and more specifically to use of encryption and hash functions to ensure security in a computerized wagering game system.

Notice of Co-pending Applications

This application is related to co-pending application 09/405,921, filed September 24, 1999, and to co-pending application 09/520,405, filed on March 8, 2000, entitled Computerized Gaming System, Method and Apparatus, which are hereby incorporated by reference.

Background of the Invention

Games of chance have been enjoyed by people for thousands of years and have enjoyed increased and widespread popularity in recent times. As with most forms of entertainment, players enjoy playing a wide variety of games and new games. Playing new games adds to the excitement of "gaming." As is well known in the art and as used herein, the term "gaming" and "gaming devices" are used to indicate that some form of wagering is involved, and that players must make wagers of value, whether actual currency or some equivalent of value, e.g., token or credit.

One popular game of chance is the slot machine. Conventionally, a slot machine is configured for a player to wager something of value, e.g., currency, house token, established credit or other representation of currency or credit. After the wager has been made, the player activates the slot machine to cause a random event to occur. The player wagers that particular random events will occur that will return value to the player. A

standard device causes a plurality of reels to spin and ultimately stop, displaying a random combination of some form of indicia, for example, numbers or symbols. If this display contains one of a preselected plurality of winning combinations, the machine releases money into a payout chute or increments a credit meter by the amount won by the player. For example, if a player initially wagered two coins of a specific denomination and that player achieved a payout, that player may receive the same number or multiples of the wager amount in coins of the same denomination as wagered.

There are many different formats for generating the random display of events that can occur to determine payouts in wagering devices. The standard or original format was the use of three reels with symbols distributed over the face of the wheel. When the three reels were spun, they would eventually each stop in turn, displaying a combination of three symbols (e.g., with three wheels and the use of a single payout line as a row in the middle of the area where the symbols are displayed.) By appropriately distributing and varying the symbols on each of the reels, the random occurrence of predetermined winning combinations can be provided in mathematically predetermined probabilities. By clearly providing for specific probabilities for each of the preselected winning outcomes, precise odds that would control the amount of the payout for any particular combination and the percentage return on wagers for the house could be readily controlled.

Other formats of gaming apparatus that have developed in a progression from the pure slot machine with three reels have dramatically increased with the development of video gaming apparatus. Rather than have only mechanical elements such as wheels or reels that turn and stop to randomly display symbols, video gaming apparatus and the rapidly increasing sophistication in hardware and software have enabled an explosion of new and exciting gaming apparatus. The earlier video apparatus merely imitated or simulated the mechanical slot games in the belief that players would want to play only the same games. Early video games therefore were simulated slot machines. The use of video gaming apparatus to play new games such as draw poker and Keno broke the ground for the realization that there were many untapped formats for gaming apparatus.

Now casinos may have hundreds of different types of gaming apparatus with an equal number of significant differences in play. The apparatus may vary from traditional three reel slot machines with a single payout line, video simulations of three reel video slot machines, to five reel, five column simulated slot machines with a choice of twenty or more distinct pay lines, including randomly placed lines, scatter pays, or single image payouts. In addition to the variation in formats for the play of games, bonus plays, bonus awards, and progressive jackpots have been introduced with great success. The bonuses may be associated with the play of games that are quite distinct from the play of the original game, such as the video display of a horse race with bets on the individual horses randomly assigned to players that qualify for a bonus, the spinning of a random wheel with fixed amounts of a bonus payout on the wheel (or simulation thereof), or attempting to select a random card that is of higher value than a card exposed on behalf of a virtual dealer.

Examples of such gaming apparatus with a distinct bonus feature includes U.S. Patent Nos. 5,823,874; 5,848,932; 5,836,041; U.K. Patent Nos. 2 201 821 A; 2 202 984 A; and 2 072 395A; and German Patent DE 40 14 477 A1. Each of these patents differ in fairly subtle ways as to the manner in which the bonus round is played. British patent 2 201 821 A and DE 37 00 861 A1 describe a gaming apparatus in which after a winning outcome is first achieved in a reel-type gaming segment, a second segment is engaged to determine the amount of money or extra games awarded. The second segment gaming play involves a spinning wheel with awards listed thereon (e.g., the number of coins or number of extra plays) and a spinning arrow that will point to segments of the wheel with the values of the awards thereon. A player will press a stop button and the arrow will point to one of the values. The specification indicates both that there is a level of skill possibly involved in the stopping of the wheel and the arrow(s), and also that an associated computer operates the random selection of the rotatable numbers and determines the results in the additional winning game, which indicates some level of random selection in the second gaming segment.

U.S. Patents Nos. 5,823,874 and 5,848,932 describe a gaming device comprising: a first, standard gaming unit for displaying a randomly selected combination of indicia, said displayed indicia selected from the group consisting of reels, indicia of reels, indicia of playing cards, and combinations thereof; means for generating at least one signal corresponding to at least one select display of indicia by said first, standard gaming unit; means for providing at least one discernible indicia of a mechanical bonus indicator, said discernible indicia indicating at least one of a plurality of possible bonuses, wherein said providing means is operatively connected to said first, standard gaming unit and becomes actuatable in response to said signal. In effect, the second gaming event simulates a mechanical bonus indicator such as a roulette wheel or wheel with a pointing element.

A video terminal is another form of gaming device. Video terminals operate in the same manner as conventional slot or video machines except that a redemption ticket is issued rather than an immediate payout being dispensed.

The vast array of electronic video gaming apparatus that is commercially available is not standardized within the industry or necessarily even within the commercial line of apparatus available from a single manufacturer. One of the reasons for this lack of uniformity or standardization is the fact that the operating systems that have been used to date in the industry are primitive. As a result, the programmer must often create code for each and every function performed by each individual apparatus. To date, no manufacturer is known to have been successful in creating a universal operating system for converting existing equipment (that includes features such as reusable modules of code) at least in part because of the limitations in utility and compatibility of the operating systems in use. When new games are created, new hardware and software is typically created from the ground up.

At least one attempt has been made to create a universal gaming engine that segregates the code associated with random number generation and algorithms applied to the random number string from the balance of the code. Carlson U.S. Patent 5,707,286 describes such a device. This patentee recognized that modular code would be beneficial, but only contemplated making RNJ and transfer algorithms modular.

The lack of a standard operating system has contributed to maintaining an artificially high price for the systems in the market. The use of unique hardware interfaces in the various manufactured video gaming systems is a contributing factor. The different hardware, the different access codes, the different pin couplings, the different harnesses for coupling of pins, the different functions provided from the various pins, and the other various and different configurations within the systems has prevented any standard from developing within the technical field. This is advantageous to the apparatus manufacturer, because the games for each system are provided exclusively by a single manufacturer, and the entire systems can be readily obsoleted, so that the market will have to purchase a complete unit rather than merely replacement software. Also, competitors cannot easily provide a single game that can be played on different hardware. A solution to this problem is presented in our co-pending application for Video Gaming Apparatus for Wagering with Universal Computerized Controller and I/O Interface for Unique Architecture, assigned serial number 09/405,921, and filed September 24, 1999, the disclosure that is incorporated herein by reference.

The invention of computerized gaming systems that include a common or universal video wagering game controller that can be installed in a broad range of video gaming apparatus without substantial modification to the game controller has made possible the standardization of many components and of corresponding gaming software within gaming systems. Such systems desirably will have functions and features that are specifically tailored to the unique demands of supporting a variety of games and gaming apparatus types, and will do so in a manner that is efficient, secure, and cost-effective.

In addition to making communication between a universal operating system and non-standard machine devices such as coin hoppers, monitors, bill validators and the like possible, it would be desirable to provide security features that enable the operating system to verify that game code and other data has not changed during operation.

Alcorn et al. U.S. Patent 5,643,086 describes a gaming system that is capable of authenticating an application or game program stored on a mass media device such as a CD-ROM, RAM, ROM or other device using hashing and encryption techniques. The

mass storage device may be located in the gaming machine, or may be external to the gaming machine. This verification technique therefore will not detect any changes that occur in the code that is executing because it tests the code residing in mass storage prior to loading into RAM. The authenticating system relies on the use of a digital signature and suggests hashing of the entire data set during the encryption and decryption process. See also, Alcorn et al. U.S. Patent 6,106,396 and Alcorn et al. U.S. Patent 6,149,522.

What is desired is an architecture and method providing a gaming-specific platform that features secure storage and verification of game code and other data, provides the ability to securely change game code on computerized wagering gaming system, and has the ability to verify that the code has not changed during operation of the gaming machine.

It is further desired that the game program code be identifiable as certified or approved, such as by the Nevada Gaming Regulations Commission or other regulatory agency.

Summary of the Invention

The invention provides an architecture and method for a wagering game-specific platform that features secure storage and verification of game code and other data, provides the ability to securely exchange data with a computerized wagering gaming system, and does so in a manner that is straightforward and easy to manage. Some embodiments of the invention provide the ability to identify game program code as certified or approved, such as by the Nevada Gaming Regulations Commission or other regulatory agency. The invention provides these and other functions by use of encryption, including digital signatures and hash functions as well as other encryption methods. Such functions are advantageously applied to data loaded into RAM and occur while the gaming machine is in operation.

In one embodiment the present invention provides a method of preparing a game data set capable of authentication. The method includes providing a game data set. A

message authentication code unique to the game data set is determined. The game data set and the message authentication code are stored.

In another embodiment, the present invention provides a method of authenticating a game used in a gaming system. The method includes receiving an encrypted control file. The encrypted control file is decrypted to provide a control file. The control file includes a set of program files, a set of message authentication codes including a message authentication code unique to each program file, and a message authentication code key. The original control file is used to verify authentication of the game.

In another embodiment, the present invention provides a gaming system. The gaming system includes a nonvolatile memory. A control file is stored in the nonvolatile memory. The control file includes a game data set, a message authentication code unique to the game data set, and a message authentication code key. A game controller is provided, wherein the game controller operates to selectively authenticate the game data set during operation of the gaming system.

In another embodiment, the present invention provides a gaming system. The gaming system includes a nonvolatile memory. An encrypted control file is stored in the nonvolatile memory. The encrypted controller file includes a set of program files, a message authentication code unique to each program file, and a message authentication code key. A gaming control is provided, wherein the gaming controller operates to decrypt the encrypted control file and authenticate the gaming program files during operation of the gaming system. Gaming system devices are provided in communication with the gaming controller via a gaming system interface.

Brief Description of the Figures

Figure 1 shows a computerized wagering game apparatus such as may be used to practice some embodiments of the present invention.

Figure 2 shows a diagram of a networked computer connected to certain components comprising a portion of a computerized wagering game apparatus, consistent with some embodiments of the present invention.

Figure 3 is a diagram of a process of creating a signature for a loadable data set, utilizing a public/private key algorithm.

Figure 4 is a diagram of a process for verifying a loadable data set has not changed during operation of the gaming device.

Figure 5 is a block diagram illustrating one exemplary embodiment of a gaming system according to the present invention.

Figure 6 is a diagram illustrating one exemplary embodiment of a process for preparing a game data set capable of authentication according to the present invention.

Figure 7 is a diagram illustrating one exemplary embodiment of a game data set and key used in a gaming system according to the present invention.

Figure 8 is a diagram illustrating one exemplary embodiment of a message authentication code process used in a gaming system according to the present invention.

Figure 9 is a diagram illustrating one exemplary embodiment of a control file used in a gaming system according to the present invention.

Figure 10 is a diagram illustrating one exemplary embodiment of a process for encrypting a control file for use in a gaming system according to the present invention.

Figure 11 is a diagram illustrating one exemplary embodiment of a process for authenticating a game used in a gaming system according to the present invention.

Figure 12 is a diagram illustrating one exemplary embodiment of a process for verifying a game program in a gaming system according to the present invention.

Detailed Description

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific sample embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, and other changes may be made without departing from the spirit or scope of the present invention. The following detailed

description is, therefore, not to be taken in a limiting sense, and the scope of the invention is defined only by the appended claims.

The present invention in various embodiments provides an architecture and method for a universal operating system that features secure storage and verification of game code and other data, provides the ability to securely exchange data with a computerized wagering gaming system, and does so in a manner that is straightforward and easy to manage. Some embodiments of the invention provide the ability to identify game program code as certified or approved, such as by the Nevada Gaming Commission or other regulatory agency. The invention provides these and other functions by use of encryption, including digital signatures and hash functions as well as other encryption methods to data being executed. Because hash functions and other encryption methods are employed widely in the present invention, they are introduced and discussed here.

"Hash functions" for purposes of this disclosure are a type of function that generates a unique data string, typically of fixed length from variable strings of characters or text. The data string generated is typically substantially smaller than the text string itself, but is long enough that it is unlikely that the same number will be produced by the hash function from different strings of text. The formula employed in the hash function must also be chosen such that it is unlikely that different text strings will produce the same hash value. An example of a suitable hash function is a 160 bit SHA hash. Regardless of file size, the hash value will be 160 bits in length.

The hashed data string is commonly referred to as a "message digest." A message digest can be stored for future use, or encrypted and then stored in nonvolatile memory, for example.

Hash functions are often used to hash data records to produce unique numeric values corresponding to each data record in a database, which can then be applied to a search string to reproduce the hash value. The hash value can then be used as an index key, eliminating the need to search an entire database for the requested data. Some hash functions are known as one-way hash functions, meaning that with such a function it is extremely difficult to derive a text string that will produce a given hash value, but

relatively easy to produce a hash value from a text string. This ensures that it is not feasible to modify the content of the text string and produce the same hash value.

Such a function can be used to hash a given character string and produce a first hash value that can later be compared to a second hash value derived from the same character string, to ensure the character string has not changed. If the character string has been altered, the hash values produced by the same hash function will be different. The integrity of the first hash value can be protected against alteration by use of other encryption methods such as the use of a digital signature.

Digital signatures are employed to sign electronic documents or character strings, and ensure that the character string has not been altered since signing. Digital signatures typically are employed to indicate that a character string was intentionally signed with an unforgeable signature that is not reusable with another document, and that the signed document is unalterable. The digital signing mechanism or method is designed to meet these criteria, typically by using complex mathematical encryption techniques.

One example is use of a public key/private key encryption system to sign a document. In a public key/private key system a user has a pair of keys, either of which may be used to encrypt or decrypt a document. The public key is published or distributed in a manner that reasonably ensures that the key in face belongs to the key owner, and the private key is kept strictly secret. If someone wishes to send a character string that only a certain person may read, the character string is encrypted before sending using the intended reader's public key. The character string is then visible only by using the intended reader's private key to decrypt the character string.

However, if a user wishes to send a character string in such a manner that the document is virtually guaranteed to be the authentic document created by the sender but essentially anyone can read it, the user can sign the document by encrypting it with his private key before sending. Anyone can then decrypt the document with the signer's public key which is typically widely distributed, and can thereby verify that the character string was signed by the key pair owner. This example embodiment meets the requirements of a digital signature, ensuring that a character string was intentionally

signed with an unforgeable signature that is not reusable with another document, and that the signed document is unalterable.

Because encryption of large character strings such as large computer programs or long text documents can require a substantial amount of time to encrypt and decrypt, some embodiments of digital signatures implement one-way hash functions. In one such embodiment, the signer uses a known one-way hash algorithm to create a hash value for the character string, and encrypts the hash value with his private key. The document and signed hash value are then sent to the recipient, who runs the same hash function on the character string and compares the resulting hash value with the hash value produced by decrypting the signed hash value with the signer's public key. Such a method provides very good security, as long as the hash function and encryption algorithm employed are suitably strong.

Encryption of data via a public key/private key system is useful not only for producing digital signatures, but also for encryption of data before sending or storing the data or to keep data secure or secret in other applications. Similarly, symmetric encryption techniques which rely on encryption and decryption of the same single secret key may be applied to such applications. For example, transmission of program data between a network server and a computerized wagering game apparatus may be secured via a symmetric encryption technique, and the program data received in the game apparatus may be verified as approved by a regulatory agency via a digital signature employing hash functions and public key cryptography before execution.

Other encryption methods and formulas exist, and are also usable consistent with the present invention. Some symmetric encryption methods, such as DES (Data Encryption Standard) and its variants rely on the secrecy of a single key, and so may not be adaptable to methods described herein that require a key pair with a public key. A variety of other encryption methods, such as RSA and Diffie-Hellman are consistent with public/private key methods, and are usable in these methods. Various hash functions may also be employed, such as MD5 or SHA, and will be useful in many aspects consistent with the present invention so long as they are sufficiently nonreversible to be

considered one-way hash functions. Various encryption methods will also provide varying degrees of security, from those that are relatively easy to defeat to those that are extremely difficult to defeat. These various degrees of security are to be considered within the scope of encryption methods consistent with this application, including various degrees of security that may to varying degrees of probability make encrypted data unforgeable, unreadable, or the like. A variety of encryption methods exist and are expected to be developed in the future, all of which are likely to be employable in some aspect consistent with the present invention, and are within the scope of the invention.

Figure 1 shows an exemplary gaming system 100, illustrating a variety of components typically found in gaming systems and how they may be used in accordance with the present invention. User interface devices in this gaming system include push buttons 101, joystick 102, and pull arm 103. Credit for wagering may be established via coin or token slot 104, a device 105 such as a bill receiver or card reader, or any other credit input device. A card reader 105 may also provide the ability to record credit information on a user's card when the user has completed gaming, or credit may be returned via a coin tray 106 or other credit return device. Information is provided to the user by devices such as video screen 107, which may be a cathode ray tube (CRT), liquid crystal display (LCD) panel, plasma display, light-emitting diode (LED) display, or other display device that produces a visual image under control of the computerized game controller. Also, buttons 101 may be illuminated to indicate what buttons may be used to provide valid input to the game system at any point in the game. Still other lights or other visual indicators may be provided to indicate game information or for other purposes such as to attract the attention of prospective game users. Sound is provided via speakers 108, and also may be used to indicate game status, to attract prospective game users, or for other purposes, under the control of the computerized game controller.

The gaming system 100 further comprises a computerized game controller 111 and I/O interface 112, connected via a wiring harness 113. The universal game controller 111 need not have its software or hardware designed to conform to the interface requirements of various gaming system user interface assemblies, but can be designed

once and can control various gaming systems via I/O interfaces 112 designed to properly interface an input and/or output of the universal computerized game controller to the interface assemblies found within the various gaming systems.

In some embodiments, the universal game controller 111 is a standard IBM Personal Computer-compatible (PC compatible) computer. Still other embodiments of a universal game controller comprise general purpose computer systems such as embedded controller boards or modular computer systems. Examples of such embodiments include a PC compatible computer with a PC/104 bus, which is an example of a modular computer system that features a compact size and low power consumption while retaining PC software and hardware compatibility. The universal game controller provides all functions necessary to implement a wide variety of games by loading various program code on the universal controller, thereby providing a common platform for game development and delivery to customers for use in a variety of gaming systems. Other universal computerized game controllers consistent with the present invention may include any general-purpose computers that are capable of supporting a variety of gaming system software, such as universal controllers optimized for cost effectiveness in gaming applications or that contain other special-purpose elements yet retain the ability to load and execute a variety of gaming software.

In yet other embodiments, the universal controller with security features can be used for other applications, including controlling networked in-line systems such as progressive controllers and player tracking systems. The invention can also be used for kiosk displays and creating picture in picture features on a video display.

The universal computerized game controller of some embodiments is a computer running an operating system with a gaming application-specific kernel such as a customized Linux kernel. In further embodiments, a system handler application layer of code executes within the kernel, further providing common game functionality to the programmer. The game program in such embodiments is therefore only a fraction of the total code, and relies on the system handler application layer and kernel to provide commonly used gaming functions. Still other embodiments will have various levels of

application code, ranging from embodiments containing several layers of game-specific code to a single-layer of game software running without an operating system or kernel but providing its own computer system management capability.

Figure 2 illustrates a networked computer connected to selected items that comprise a part of a computerized wagering game apparatus, as are used in various embodiments of the present invention. The computerized game controller 201 has a processor 202, memory 203, and nonvolatile memory 204. One example of nonvolatile memory is a flash disk on chip (hereinafter "flash disk"). The flash disk is advantageously read/write, yet retains information stored on disk upon power down. Attached to the computerized game controller of some embodiments is a mass storage device 205, and a network interface adaptor 206. The network interface adaptor is attached to a networked computer 207 via network connection 208. The various components of Figure 2 exist within embodiments of the invention, and are illustrated to show the manner in which the various components are associated.

The computerized wagering game controller of the invention is operable to control a computerized wagering game, and is operable to employ encryption in various embodiments to provide data security. The computerized game controller 201 in some embodiments is a general-purpose computer, such as an IBM PC-compatible computer. The game controller executes an operating system, such as Linux or Microsoft Windows, which in further embodiments is modified to execute within the computerized gaming apparatus. The computerized game controller also executes game code, which may be loaded into memory 203 from either a mass storage device 205 such as a hard disc drive, or nonvolatile memory 204 such as flash memory or EPROM memory before execution. In some embodiments, the computerized game controller 201 loads encryption functions into memory 203, and those functions are subsequently executed to securely load other gaming system data from the mass storage device 205.

In further embodiments, the computerized game controller exchanges data with a networked computer 207 via a network connection 208 and a network interface adapter 206. Data exchanged via the network connection is encrypted in some embodiments of

the invention, to ensure security of the exchanged data. The data to be exchanged in various embodiments comprises game program data, computerized gaming apparatus report data, data comprising commands to control the operation of the computerized gaming apparatus, and other computerized gaming apparatus data. Employing encryption in exchanging such data provides a degree of security, ensuring that such data is not altered or forged.

The invention employs encryption, including hash functions, symmetric encryption, and public key/private key encryption in various embodiments, which provides a degree of confidence that data utilized by the computerized gaming system and protected by encryption in accordance with the invention is not altered or forged. The data within the scope of the invention includes but is not limited to data comprising programs such as operating system or game program data, computerized gaming machine status data such as credits or other game state data, control instruction data for controlling the operation of the computerized gaming apparatus, and other computerized gaming machine data.

One embodiment of the invention comprises the use of hash functions to calculate a reference hash value for selected data, which can later be compared to a hash value calculated from the same data or a copy of the data to ensure the data has not been altered. The hash functions employed will desirably be one-way hash functions, to provide a greater degree of certainty that the reference hash value cannot be used in reverse to produce corresponding altered data. In a further embodiment, the data is hashed repeatedly by a continuously executing program thread that ensures that the data is not altered during the course of operation of the computerized wagering game. The data that is continuously hashed is in some embodiments is continuously hashed after being loaded into memory 203 for use by the computerized game controller.

If the reference hash value and the calculated hash value do not match, the computerized gaming apparatus will desirably provide some indication of the hash failure. In one embodiment, the game is brought to a locked or "tilt" state that prevents wagering upon a hash check failure. In a further embodiment, notification of the hash

failure is sent to a networked computer 207 to alert the computer's user of the hash failure. In some embodiments, the computerized wagering game apparatus provides limited function to check the status of the game, including in further embodiments functions accessible only by operating controls within the computerized wagering game apparatus secure housing.

In one embodiment, the operating system as described in my copending application for Computerized Gaming System, Method and Apparatus, having Serial Number 09/520,405 and filed on the March 8, 2000, cooperates with a library of "shared objects" that are specific to the game application. For purposes of this disclosure, a "shared object" is defined as self-contained, functional units of game code that define a particular feature set or sequence of operation for a game. The personality and behavior of a gaming machine of the present invention are defined by the particular set of shared objects called and executed by the operating system. Within a single game, numerous shared objects may be dynamically loaded and executed. This definition is in contrast with the conventional meaning of a shared object, which typically provides an API to multiple programs. An API is an application Programming Interface, and includes a library of functions.

The shared object code, as well as other data may be verified according to one embodiment of the present invention by first preparing a signature from data, as shown in Figure 3. The signature may be prepared by first hashing 210 the data set 212 to create a message digest 214. The message digest is encrypted via an encryption program that is stored on ROM utilizing a private/public key algorithm 218, forming a unique signature 220. The data and signature are then stored on a mass storage device 222 such as a network storage device, hard drive, CD-ROM, RAM, flash disk or the like.

In one embodiment, the shared objects for a particular application and their corresponding signatures are stored 224 in flash memory. When the shared objects are called, it is copied into RAM, where it is hashed 226 on a frequent periodic basis. The shared objects may be hashed from flash memory, or loaded into RAM and then hashed from RAM. Utilizing a Linux, Unix or other similar operating system advantageously

permits the location of data in RAM. Data verification in RAM has the distinct advantage that errors will be caught at the time they occur, rather than when the data is loaded or reloaded. This could save casinos untold amounts by avoiding the payment of jackpots and the like based on machine malfunction. Since hashing is a batch process, the process is not continuous. However, when the hashing takes relatively little time, such as 10 seconds for example, the process can repeat itself so that the data verification in RAM is in effect, continuous.

The message digest 228 created from hashing the shared object is preferably encrypted. A public key 238 is used to decrypt the message digest utilizing a first decryption program. The signature 240 stored in flash memory is decrypted using a second decryption program via a public key 234 and the values are compared 236.

Although code verification of the gaming program shared objects has been described in detail above, code verification utilizing hash functions and signatures can be applied to verifying the authenticity of the linux kernel, modular modifications to the kernel, the operating system, game state data, random number generation data and the like. As added security, the present invention contemplates zeroing out all unused RAM to verify that no data in the form of code or other data was intentionally or unintentionally inserted.

In various embodiments, selected data is protected with encryption by signing the data with a digital signature that is verified to ensure integrity of the data. In some embodiments, the digital signature comprises signing the selected data with a signer's private key such that the data can only be decrypted by using the corresponding public key. Because only the intended signer knows his private key and documents encrypted with other private keys cannot be decrypted with the intended signer's public key, successful decryption of data with the intended signer's public key provides a degree of certainty that the data was signed or encrypted by the intended signer.

But, because public key/private key encryption algorithms typically take a relatively long time to encrypt large amounts of data, the encryption algorithm is more efficiently used in some embodiments to encrypt a unique characteristic of the data such

as the hash value from a one-way hash function. In such an embodiment, the signer derives the reference hash value with a one-way hash function for the data to be signed, and encrypts the resulting hash value with his public key. One-way hash functions typically may be applied to data much more quickly than public key/private key algorithms, and so it is more desirable to process the entire data to be signed with a hash function than with a public key/private key algorithm. In some embodiments of the invention, only the hash value needs to be encrypted with public key/private key encryption, greatly reducing the time needed to sign or verify large amounts of data. To verify the signature, the hash value is decrypted with the intended signer's public key and the decrypted reference hash value is compared to a newly-computed hash value of the same data. If the reference hash value matches the newly-computed hash value, a degree of certainty exists that the signed data has not been altered since it was signed.

In some embodiments using digital signatures, the digital signature is that of a regulatory agency or other organization responsible for ensuring the integrity of data in computerized wagering game systems. For example, the Nevada Gaming Regulations Commission may apply a signature to data used in such gaming systems, ensuring that they have approved the signed data. Such an embodiment will be useful to ensure that game code executing in these systems has been approved and not altered since approval, and provides security both to the game operator or owner and to the regulatory commission. In other embodiments, the digital signature is that of the game code manufacturer or designer, and ensures that the game code has not been altered from its original state since signing.

Secure storage of the reference hash values or public keys in the systems described above is important, because data can be more easily forged if the reference hash values or public keys used to verify the integrity of the data can also be altered. For this reason, the reference hash values, public keys, or other encryption key data is stored in nonvolatile memory 204. In some embodiments, the nonvolatile memory 204 is a flash memory or EPROM that is programmable, but is not readily altered by a user of the computerized wagering game apparatus. The nonvolatile memory in such embodiments

is reprogrammable, but reprogramming requires in various embodiments the use of special hardware, execution of restricted functions, or other secure methods. In other embodiments, the nonvolatile memory 204 is a programmable memory that is not alterable, requiring replacement of the nonvolatile memory each time new encryption key data is needed. Such embodiments have the advantage that the nonvolatile memory 204 must be physically removed and replaced to alter the data, providing a degree of access security and allowing visual verification of the identity of the nonvolatile memory and its contents.

In still other embodiments, the encryption key data is stored on the mass storage device. Further embodiments include storage of the encryption key data embedded in encryption functions, storage in secure areas of a hard disc drive mass storage device, or use of other security methods to protect the encryption key data.

These encryption methods in some embodiments of the invention are also applied to computerized gaming system communication over a network. Data communicated over a network is in various embodiments of the invention verified by use of a hash function, verified by use of public key/private key encryption, verified by use of symmetric encryption, or verified by use of digital signatures. Also, a variety of key exchange or key negotiation protocols exist which in some embodiments of the invention provide the capability for a networked computerized gaming system to publicly agree with another networked computer system on encryption keys that may be subsequently used to communicate securely over a network.

Such network communication methods are utilized in the invention to provide for secure exchange of data between computerized wagering game systems and other networked computer systems. For example, control commands that control certain aspects of the operation of the computerized wagering games are securely sent over a network in some embodiments of the invention. Such commands may include increasing odds of payout on selected computerized wagering game systems, or changing the game program that is executed on selected computerized wagering game systems at selected times of the day. The computerized wagering games in some embodiments securely

report game data such as bookkeeping data to a networked computer 207 via encryption. In still other embodiments of the invention, wagering game program data is securely transmitted over the network to the computerized wagering game systems, providing a secure way to provide new wagering games to the systems without physically accessing each computerized wagering game system. Various embodiments of the invention transmit other computerized wagering game data over a network connection via encryption, and are within the scope of the invention.

Because encryption methods typically provide a degree of security that is dependent on the effort and expense a hacker is willing to invest in defeating the encryption, replacement of encryption keys is employed in some embodiments of the invention. Digital signatures in some embodiments are valid only for a predetermined period of time, and in further embodiments have an associated date of expiry after which they may no longer be used. Such methods can also be used in various embodiments of the invention to license games for use for a certain period of time, after which they will not be properly verified due to expiry of the encryption keys used for data verification. Because hash functions typically produce hash values that are dependent entirely on the data being hashed, embodiments of the invention which incorporate expiry and replacement of reference hash values also require reissuance of modified data to produce a different hash value. For example, minor bug fixes, addition of new features, or any other small change in the data comprising a gaming program will be sufficient to produce a different reference hash value upon hashing the edited program data, resulting in an updated reference hash value corresponding to the updated data.

Other embodiments use a variety of keys among various computerized wagering games and game producers, reducing the risk and therefore the value of successfully defeating an encryption key. For example, a game producer in one embodiment employs a different digital signature for each customer of its computerized wagering games, ensuring that defeating the encryption key on a single game system affects a limited number of games. In another embodiment, a regulatory agency may change keys with which it signs games on a periodic basis, so that a successful hack of the keys used to

sign the data results in potential compromise of only a limited and identifiable number of games. It will be obvious to one skilled in the art that many variations on key replacement and expiry policies exist, all of which are considered within the scope of the present invention.

The invention provides an architecture and method for a gaming-specific platform that features secure storage and verification of game code and other data, provides the ability to securely exchange data with a computerized wagering gaming system, and does so in a manner that is straightforward and easy to manage. Some embodiments of the invention provide the ability to identify game program code as certified or approved, such as by the Nevada Gaming Regulations Commission or other regulatory agency. The invention provides these and other functions by use of encryption, including digital signatures and hash functions as well as other encryption methods.

Figure 5 is a block diagram illustrating one exemplary embodiment of a gaming system according to the present invention. The gaming system block diagram is representative of gaming system 100 shown in Figure 1 and Figure 2, and previously described herein. The gaming system 100 includes a unique system and method for preparing a game data set capable of authentication and authenticating a game used in the gaming system 100. The gaming system 100 includes a process which securely verifies that the gaming data set, including program files have not been altered, either intentionally or unintentionally, changing the outcome of a game played on the gaming system 100.

Components of the present invention can be implemented in hardware via a microprocessor, programmable logic, or state machine, in firmware, or in software within a given device. In one preferred embodiment, one or more components of the present invention reside in software. Components of the present invention may also reside in software on one or more computer-readable mediums. The term computer-readable medium as used herein is defined to include any kind of memory, volatile or nonvolatile, such as floppy disks, hard disks, CD-ROMs, flash memory, read-only memory (ROM), and random access memory (RAM). In addition, gaming system 100 can employ a

microprocessor embedded system/appliance incorporating tailored appliance hardware and/or dedicated signal purpose hardware.

In one aspect, gaming system 100 includes a gaming control system 300, gaming system interface 302, and gaming system devices 304. Gaming control system 300 includes computer or controller 201, nonvolatile memory 204, and nonvolatile memory 306. Controller 201 includes memory 203 and nonvolatile RAM (NVRAM) 308. In one aspect, memory 203 is random access memory. In one aspect , the random access memory 203 is dynamic random access memory (DRAM). The nonvolatile random access memory includes a battery backup for maintaining data stored in memory upon loss of power. In one embodiment, NVRAM 308 is used for storing crucial gaming data, such as slot machine reel settings, payoff percentages, and credits.

In one embodiment, program memory 204 is a read/writeable, nonvolatile memory. In one aspect, the writeable memory 204 is flash memory. One suitable nonvolatile memory is commercially available under the trade name "Disk on a Chip" commercially available from M Systems. Other nonvolatile memory suitable for use with the present invention will become apparent to one skilled in the art after reading the present application.

Nonvolatile memory 24 is used to store a game data set, which is defined to include game specific code or gaming program files. Exemplary game specific codes includes game code, game data, game sound, game graphics, game configuration files, or other game specific files. The game specific code or program files are directed to specific type of games run on the gaming system, such as Blackjack, poker, video slot machines, or reel slot machines. In one embodiment, nonvolatile memory 306 is read only memory (ROM) such as an EEPROM. Nonvolatile memory 306 is used to store gaming system operating code. Upon power up or operation of the gaming system, the gaming system operating code and game data sets are transferred into memory, preferably volatile memory 203, for fast access by controller 201 for operation of the gaming system. During operation of the gaming system 100, controller 201 interfaces with gaming system devices 304 via gaming system 302 for operation of the gaming

system 100. Gaming system interface 302 may include network interface 206, network computer 207, and network connection 208 previously detailed herein. Gaming system devices 304 include mechanical, electrical, hardware, software or video devices, such as pushbuttons 101, joystick 102, pull arm 103, token or slot 104, device 105, point tray 106, video screen 107 and speakers 108 previously detailed herein.

The gaming system 100 according to the present invention includes an encrypted control file 310 and associated game files stored in the nonvolatile memory 204. The encrypted control file 310 includes the game data set, such as game specific code and program filenames, message authentication codes unique to the program filenames, and a message authentication code key. A message authentication code process 312 is stored in nonvolatile memory 306. In one aspect, the control file 310 is encrypted. The control file 310 is used in connection with the message authentication code process 312 to provide game data security during operation of the gaming system 100, as part of a game authentication/verification process. The game authentication/verification process is described in detail in reference to the following Figures 6-11.

Figure 6 is a diagram illustrating one exemplary embodiment of a method of preparing a game data set capable of authentication. A game data set is indicated at 320. As indicated herein the game data set 20 includes game specific code filenames or program filenames for game files, such as game code, game data, game sound, game graphics, game configuration files, and other game specific files. A message authentication code is determined which is unique to the game data set 320 but is determined using less than the whole game data set (i.e., the whole data set being the program file and program filenames). The message authentication code is determined using a message authentication code process 322 (MAC process). In one aspect, the message authentication codes are determined using the filenames associated with the program files, resulting in fast determination of the unique message authentication codes. The term message authentication code as used herein, also known as a data authentication code, is a one-way hash function with the addition of a secret key, indicted as message authentication code key 324. A resultant hash value is a function of both the

pre-image game data set 320 and the message authentication code key 324. See, Applied Cryptography, 1996 Second Edition, by Bruce Schneier, Chapter 18 which is incorporated herein by reference. The output of the message authentication code process 322 is stored. In one aspect, the game data set, the message authentication code, and the message authentication code key are stored in a control file 326 in memory.

Figure 7 is a diagram illustrating one exemplary embodiment of game data set 320 and message authentication code key 324. In one aspect, game data set 320 includes a plurality of game specific code or program filenames, indicated as FILENAME1 328, FILENAME2 330, through FILENAMEN 332.

Figure 8 is a diagram illustrating one exemplary embodiment of a message authentication code process 322 used in the present invention, including being used in preparing a game data set capable of authentication for a gaming system according to the present invention. In this embodiment, the message authentication code process utilizes a public-key encryption algorithms in a block chaining mode as a one-way hash function. Game data set 320 includes program filenames FILENAME1 328, FILENAME2 330 through FILENAMEN 332. A message authentication code is determined which is unique to each program filename FILENAME1 328, FILENAME2 330 through FILENAMEN 332. A message authentication code function 334 is defined for the message authentication code process 322. Program FILENAME1 328 and message authentication code key 324 are applied to the message authentication code function to determine message authentication code 336 (MAC1). Utilizing a block chaining scheme, the message authentication code MAC1 336 is used as the "key" for determining the next message authentication code unique to the next filename. As such, the validity of the message authentication code process 322 is also dependent on the order in which the message authentication codes are determined, and the validity of the message authentication code output from each previous step.

Program FILENAME2 330 and the message authentication code MAC1 336 are applied to message authentication code function 334 to determine message authentication code MAC2 338. This process is continued for each subsequent program filename. As

such, program FILENAMEN 332 and the last determined message authentication code are applied to message authentication code function 334 to determine the message authentication code FILENAMEN 340.

For increased security, a message authentication code is again determined for the program filename FILENAME1 utilizing the last determined message authentication code. FILENAME 328 and message authentication code MACN 340 are applied to message authentication code function 334 to provide a message authentication code MAC1X or (MAC1' 342). In this embodiment, each message authentication code unique to each program filename is dependent upon a previously determined message authentication code. Deferring the message authentication code using each filename is much faster than hashing entire program files in an authentication scheme requiring hashing, and the subsequent determination of digital signatures using an encryption scheme.

Figure 9 is a diagram illustrating one exemplary embodiment of control file 326 generated after completion of the message authentication code process 322. Control file 326 includes each program filename in the game data set 320, including FILENAME1 328, program FILENAME2 330 through program FILENAMEN 332. Control file 326 also includes the message authentication code key 324, and the unique message authentication code unique to each program file. In particular, message authentication code MAC1 unique to FILENAME1, also message authentication code MAC1X 336 which is unique to program FILENAME1 328, message authentication code MAC2 338 which is unique to program FILENAME2 330, through message authentication code MACN 340 which is unique to program FILENAMEN 332.

Figure 10 is a block diagram illustrating one exemplary embodiment of a process for providing a secure gaming system according to the present invention. In one aspect, control file 326 is encrypted using encryption program 350, to provide an encrypted control file 352. The encrypted control file 352 is stored in program memory, indicated at 354. In reference also to Figure 5, the encrypted control file is shown stored in nonvolatile memory 204 as control file 310 for use by gaming system 100. Additionally,

the program files associated with the encrypted control file are also stored in memory 204.

In one aspect, encryption program 350 utilizes a private key 356 and a public key 358 as part of a public key/private key encryption process similar to the public key/private key encryption process previously described herein. One encryption process suitable for use as encryption program 350 in the present invention utilizes an ElGamal encryption scheme. Other encryption methods may be utilized which may or may not use public key/private key encryption systems, such as RSA and Diffie-Hellman, may be employed. Various hash functions may also be employed, such as MD5 or SHA. Preferably, the hash functions are one-way hash functions.

Figure 11 is a diagram illustrating one exemplary embodiment of a method of authenticating a game used in a gaming system 100 according to the present invention. Reference is also made to Figures 1-10 previously detailed herein. The game can be verified as authentic at selected times, such as during game power-up, or when game data, including game program files, is transferred from nonvolatile memory 204 for use by the gaming system 100. Further, once transferred into RAM 203, the authentication of the game data set or game program files can be checked at (continuously or at desired intervals) during operation of the game to verify authentication of the game.

In one aspect, encrypted control file 352 is received from nonvolatile memory 204 and decrypted using a corresponding decryption program 360. In one aspect, decryption program 360 utilizes public key 358. The decryption program 360 reverses the encryption provided by encryption program 350. The application of decryption program 360 to encrypted control file 352 results in the original control file 326. Control file 326 includes the game data set 320, having program filenames FILENAME 1, FILENAME 2 through FILENAMEN. Control file 326 further includes the corresponding unique message authentication codes MAC1, MAC2 through MACN, and MAC1X and message authentication code key 324.

The game program files are compared with the previously determined message authentication codes in order to verify authenticity of the game and in particular the

game programs . The program filenames and message authentication code key are applied to the same message authentication code process 322, as previously detailed in Figure 8, providing an output of complimentary message authentication codes 362. At 364, the message authentication codes from control file 326 are compared to the corresponding determined complimentary message authentication codes 362. As indicated at 366, if the message authentication codes and the complimentary message authentication codes set match, the game is verified authentic and use of the game programs is allowed to continue, indicated at 368. If the message authentication codes and the complimentary message authentication codes do not match, the game is not verified as authentic and enters an error mode, is terminated and/or system operating personnel are notified, indicated at 370.

In Figure 12, one exemplary embodiment of a game verification process used in a gaming system according to the present invention is generally shown at 380. In verification process 380, after the game data set 382 has been authenticated and transferred into RAM 203, the present invention provides for continuous verification of the game data set to assure that the game data set 382 has not changed from the original game data set stored in nonvolatile memory 204. In particular, a hash function 384 is applied to the game data set 382, resulting in a hashed output stored in message digest 386. Message digest 386 comprises a unique hashed output corresponding to each program file in game data set 382. In one aspect, hash function 384 is a SHA hash function. Other suitable hash functions include MD5, SNEFRU, HAVAL and N-HASH. Other hash functions which are suitable for use in the verification process according to the present invention will become apparent to one skilled in the art after reading the present application. The hashed output or message digest 386 is stored in a storage system 388. The storage system 388 may include message digest 386 being stored in RAM 203 or in VRAM 308 or other suitable storage system which is part of gaming system 100.

During operation of the gaming system, the gaming data set 382 may be continuously verified to determine that no change has occurred in the game data set. In

one aspect, the game data set 382 is verified one file at a time. In particular, during operation of the gaming system, a program file is applied to hash function 390, wherein hash function 390 is the same as hash function 394. At 392, the hashed output of hash function 390 is compared to the corresponding hashed output stored at system 388. At 394, if no match occurs the game enters into an error mode, is terminated, and/or gaming personnel are notified, indicated at 396. At 398, if a match occurs the next program file of game data set 382 is verified in a similar manner. As such, the game data set 382 is continuously verified during operation of the gaming system. Another aspect, the game data set may be verified using the verification process according to the present invention at desired time intervals or upon the occurrence of a desired event, such as the start of each game played on the gaming system.

The gaming system 100 according to the present invention provides a unique system and method for preparing a game data set capable of authentication and authenticating a game used in the gaming system 100. The gaming system 100 includes a process which securely verifies that the gaming set, including program files have not been altered, either intentionally or unintentionally, which could result in the changing of the outcome of a game played on the gaming system 100. In one aspect, the present invention provides for continuous verification of the gaming system 100 during operation of the gaming system 100.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the invention. It is intended that this invention be limited only by the claims, and the full scope of equivalents thereof.

What is claimed is:

1. A computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and nonvolatile storage and operable to control the computerized wagering game; and

game data stored in the nonvolatile storage, wherein the game data stored in nonvolatile storage is verified during operation.

2. The computerized wagering game apparatus of claim 1, wherein the game data securely stored in the nonvolatile storage is hashed with a one-way hash function and a resulting hash value is compared to a reference hash value to ensure that the gaming program has not changed since calculation of the reference hash value.

3. The computerized wagering apparatus of claim 2, wherein the game data is hashed after loading the gaming program into random access memory and the resulting hash value is compared to a reference hash value in a continuously executing program thread executing on the computerized game controller.

4. The computerized wagering game apparatus of claim 2, wherein the computerized wagering game system is brought to a tilt state if the resulting hash value is not the same as the reference hash value.

5. The computerized wagering game apparatus of claim 2, wherein the reference hash value is stored in a nonvolatile memory comprising a part of the computerized wagering game apparatus.

6. The computerized wagering game system of claim 1, wherein a system handler application loads and executes encryption functions which are subsequently used to securely load other game data from nonvolatile storage.

7. The computerized wagering game apparatus of claim 1, wherein the game data securely stored in the nonvolatile storage via encryption is signed with a digital signature.

8. The computerized wagering game apparatus of claim 7, wherein the digital signature comprises encryption of the gaming program data with a signer's private key.

9. The computerized wagering game apparatus of claim 8, further comprising a nonvolatile memory storing a public key corresponding to the signer's private key.

10. The computerized wagering game apparatus of claim 7, wherein the digital signature comprises encryption with a signer's private key of a hash value produced by hashing the gaming program data with a one-way hash function.

11. The computerized wagering game apparatus of claim 10, further comprising a nonvolatile memory storing a public key corresponding to the signer's private key.

12. The computerized wagering game apparatus of claim 7, wherein the gaming program data signed with a digital signature is signed with a digital signature from a regulatory organization, thereby signifying organization approval of the gaming program data.

13. The computerized wagering game apparatus of claim 1, wherein the computerized game controller is a general-purpose computer.

14. The computerized wagering game apparatus of claim 12, wherein the general-purpose computer is an IBM PC-compatible computer.

15. The computerized wagering game apparatus of claim 1, further comprising a network interface connecting the computerized wagering game apparatus to a networked computer.

16. A method for securing data on a computerized wagering game apparatus, comprising verification of game data located in RAM during operation of a computerized gaming apparatus.

17. The method of claim 16, further comprising encryption of data communicated via the computerized wagering game apparatus over a network.

18. The method of claim 17, wherein the data communicated over the network comprises instructions to control the operation of the computerized wagering game.

19. The method of claim 17, wherein the data communicated over the network comprises shared objects for execution on the computerized wagering game.

20. The method of claim 17, wherein the data communicated over the network comprises data reported by the computerized wagering game.

21. The method of claim 16, wherein encryption of data stored in the computerized gaming apparatus comprises:

hashing the stored data with a one-way hash function; and

comparing a resulting hash value to a reference hash value to ensure that the data has not changed since calculation of the reference hash value.

22. The method of claim 21, wherein the reference hash value is stored in nonvolatile memory that comprises a part of the computerized wagering game apparatus.

23. The method of claim 16, wherein hashing the stored data with a one-way hash function comprises:

loading the data into random access memory;

hashing the stored data with a one-way hash function in a continuously executing thread; and

comparing a resulting hash value to a reference hash value in a continuously executing thread to ensure that the data has not changed since calculation of the reference hash value.

24. The method of claim 21, further comprising bringing the computerized wagering game to a tilt state if the resulting hash value is not the same as the reference hash value.

25. The method of claim 16, wherein encryption of data stored in the computerized gaming apparatus comprises signing the data with a digital signature.

26. The method of claim 25, wherein signing the data with a digital signature comprises encryption of the data with a signer's private key.

27. The method of claim 26, wherein a public key corresponding to the signer's private key is stored in nonvolatile memory comprising a part of the computerized wagering game apparatus.

28. The method of claim 26, wherein signing the data with a digital signature comprises:

computing a hash value from the data produced with a one-way hash function; and

encrypting the hash value with a signer's private key.

29. The method of claim 28, wherein a public key corresponding to the signer's private key is stored in nonvolatile memory comprising a part of the computerized wagering game apparatus.

30. The method of claim 16, wherein the computerized wagering game apparatus comprises a general-purpose computer.

31. The method of claim 30, wherein the general-purpose computer comprises an IBM PC-compatible computer.

32. The method of claim 16, wherein encrypting data comprises use of a symmetric encryption algorithm to encrypt data.

33. A machine-readable medium with instructions stored thereon, the instructions when executed operable to cause a computerized wagering game apparatus to:

apply encryption to data stored in the computerized gaming apparatus.

34. The machine-readable medium of claim 33, further comprising instructions that when executed are further operable to cause the computerized wagering game apparatus to:

apply encryption to data communicated via the computerized wagering game apparatus over a network.

35. The machine-readable medium of claim 33, wherein applying encryption to data stored in the computerized gaming apparatus comprises:

hashing the data with a one-way hash function; and

comparing a resulting hash value to a reference hash value to ensure that the data has not changed since calculation of the reference value.

36. The machine-readable medium of claim 33, wherein applying encryption to data stored in the computerized gaming apparatus comprises signing the data with a digital signature.

37. The machine-readable medium of claim 36, wherein signing the data with a digital signature comprises encryption of the data with a user's private key.

38. The machine-readable medium of claim 36, wherein signing the data with a digital signature comprises:

computing a hash value from the data produced with a one-way hash function; and

encrypting the hash value with a signer's private key.

39. A computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory and nonvolatile storage and operable to control the computerized wagering game;

gaming program code and gaming program code signature stored in the nonvolatile storage, and

an authentication program stored in nonvolatile storage, wherein the authentication program, when executed, verifies that the gaming program code in nonvolatile storage has not changed by means of generating a message digest from the gaming program code, decrypting the message digest using a first decryption program; decrypting the gaming program code signature with a second decryption program and comparing the two decrypted messages to verify that they are identical.

40. A method of preparing a game data set capable of authentication comprising:

providing a game data set;

determining a message authentication code unique to the game data set; and

storing the game data set and the message authentication code.

41. The method of claim 40, further comprising:

defining a key; and

using the key to determine the message authentication code unique to the game data set.

42. The method of claim 41, further comprising:

storing the key with the game data set and the message authentication code.

43. The method of claim 40, further comprising:

defining the game data set to include a set of program files, and wherein determining the message authentication code unique to the game data set includes determining a message authentication code unique to each program file in the program file set.

44. The method of claim 40, further comprising:

defining a key;

defining a message authentication code function;

determining the message authentication code unique to the game data set by applying the game data set and the key to the message authentication code function.

45. A method of claim 40, further comprising:

defining a key;

defining the game data set to include a set of program files, the set of program files including a first game program file and a second game program file;

defining a message authentication code function; and

determining a first message authentication code unique to the first game program file by applying the first game program file and the key to the message authentication code function; and

determining a second message authentication code unique to the second game program file by applying the second game program file and the first message authentication code to the message authentication code function.

46. The method of claim 45, further comprising:

defining the game data set to include N game program files;

determining an N message authentication code unique to an N program file by applying the N program file and a last determined message authentication code to the message authentication code function.

47. The method of claim 46, wherein the last determined message authentication code is the second message authentication code.

48. The method of claim 46, further comprising:

determining a first prime message authentication code for the first game program file by applying the first game program file and the N message authentication code to the message authentication code function.

49. A method of authenticating a game used in a gaming system comprising:

receiving an encrypted control file,

decrypting the encrypted control file to provide a control file, the control file including a set of program files, a set of message authentication codes including a message authentication code unique to each program file, and a message authentication code key; and

using the original control file to verify authentication of the game.

50. The method of claim 49 wherein the step of using the control file to verify authentication of the game comprises:

determining a complimentary message authentication code set including a complementary message authentication code unique to each program file in the set of program files using the set of program files and the message authentication code key; and

comparing the message authentication code set to the complimentary message authentication code set to verify authentication of the game.

51. The method of claim 50, wherein if the message authentication code set and the complimentary message authentication code set match, further comprising indicating that the game is verified authentic.

52. The method of claim 50, wherein if the message authentication code set and the complimentary message authentication code set do not match, indicating that the game is not verified authentic.

53. The method of claim 49 further comprising:

receiving the control file, including the set of program files, the set of message authentication codes including the message authentication code unique to each program file, and the message authentication code key;

encrypting the control file to provide the encrypted control file;

storing the encrypted control file.

54. The method of claim 53, further comprising using a private key to encrypt the control file.

55. The method of claim 53, further comprising using a public key to encrypt the control file.

56. The method of claim 49, further comprising decrypting the encrypted control file using a public key.

57. The method of claim 49, further comprising generating the control file including:

providing a game data set including the set of program files;

defining the message authentication code key;

determining the message authentication code unique to the game data set using the game data set and the message authentication code key, including determining the message authentication code unique to each program file; and

storing the set of program files, the message authentication codes, and the message authentication code key as the control file.

58. A gaming system comprising:

a nonvolatile memory;

a control file stored in the nonvolatile memory, the control file including a game data set, a message authentication code unique to the game data set, and a message authentication code key; and

a game controller, wherein the game controller operates to selectively authenticate the game data set using the message authentication code unique to the game data set during operation of the gaming system.

59. The system of claim 58, wherein the control file is an encrypted control file, and wherein the controller operates to decrypt the encrypted control file to provide the control file.

60. The system of claim 58, wherein the game controller include volatile memory, and wherein the game controller operates to authenticate the game data set each time the game data set is transferred from the nonvolatile memory to volatile memory for use by the gaming system.

61. A gaming system comprising:

a nonvolatile memory;

an encrypted control file stored in the nonvolatile memory, the encrypted control file including a set of program files, a message authentication code unique to each program file, and a message authentication code key;

a gaming controller, wherein the gaming controller operates to decrypt the encrypted control file and authenticate the gaming program files during operation of the gaming system; and

gaming system devices in communication with the gaming controller via a gaming system interface.

62. The system of claim 61, further comprising a message authentication code process stored in memory, wherein the game controller authenticates the set of program files by applying the message authentication process using the set of program files and the message authentication code key to provide a set of complimentary message authentication codes, and comparing the message authentication codes from the control file to the complimentary message authentication codes.

63. The system of claim 62, wherein the message authentication process is stored in read only memory.

64. The system of claim 61, wherein the nonvolatile memory is writeable memory.

65. The system of claim 64, wherein the nonvolatile memory is flash memory.

66. A computer-readable medium having computer-executable instructions for performing a method of preparing a game data set capable of authentication comprising:

providing a game data set;

determining a message authentication code unique to the game data set; and

storing the game data set and the message authentication code.

67. A computer-readable medium having computer-executable instructions for performing a method of authenticating a game used in a gaming system comprising:

receiving an encrypted control file;

decrypting the encrypted control file to provide a control file, the control file including a set of program files, a set of message authentication codes including a message authentication code unique to each program file, and a message authentication code key; and

using the original control file to verify authentication of the game.

68. A method of continuously verifying a game used in a gaming system comprising:

receiving a game data set;

determining a hashed output unique to the game data set;

storing the hashed output;

determining a complimentary hashed output unique to the game data set during operation of the game;

comparing the stored hashed output to the complimentary hashed output.

69. The method of claim 68, wherein if the stored hashed output matches the complimentary hashed output, continuing to verify the game data set.

70. The method of claim 68, wherein if the stored hashed output does not match the complimentary hashed output, terminating operation of the game.

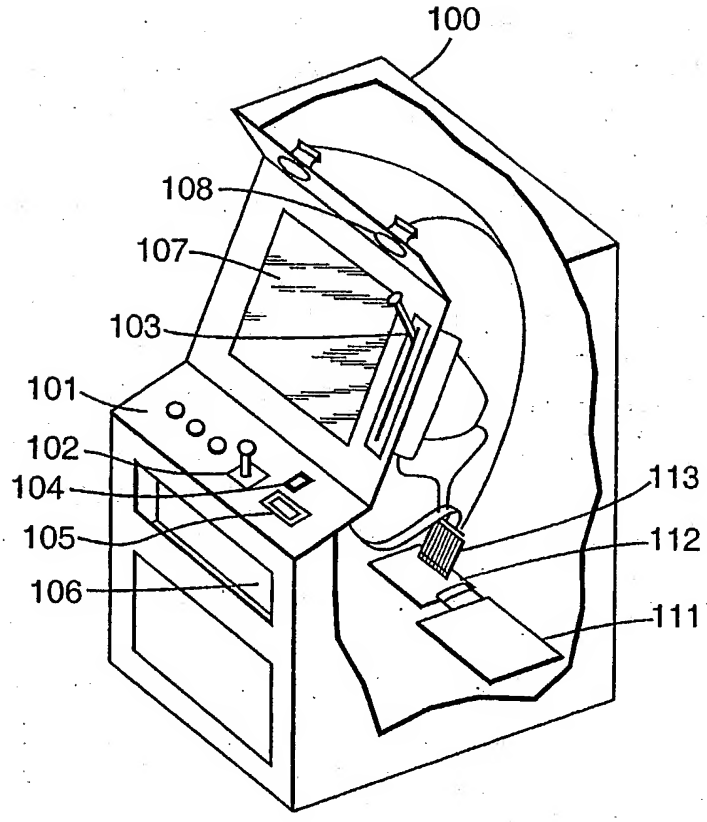

Fig. 1

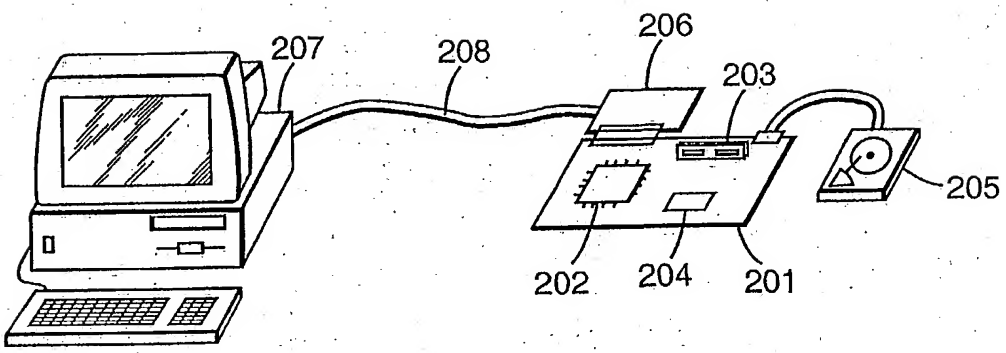

Fig. 2

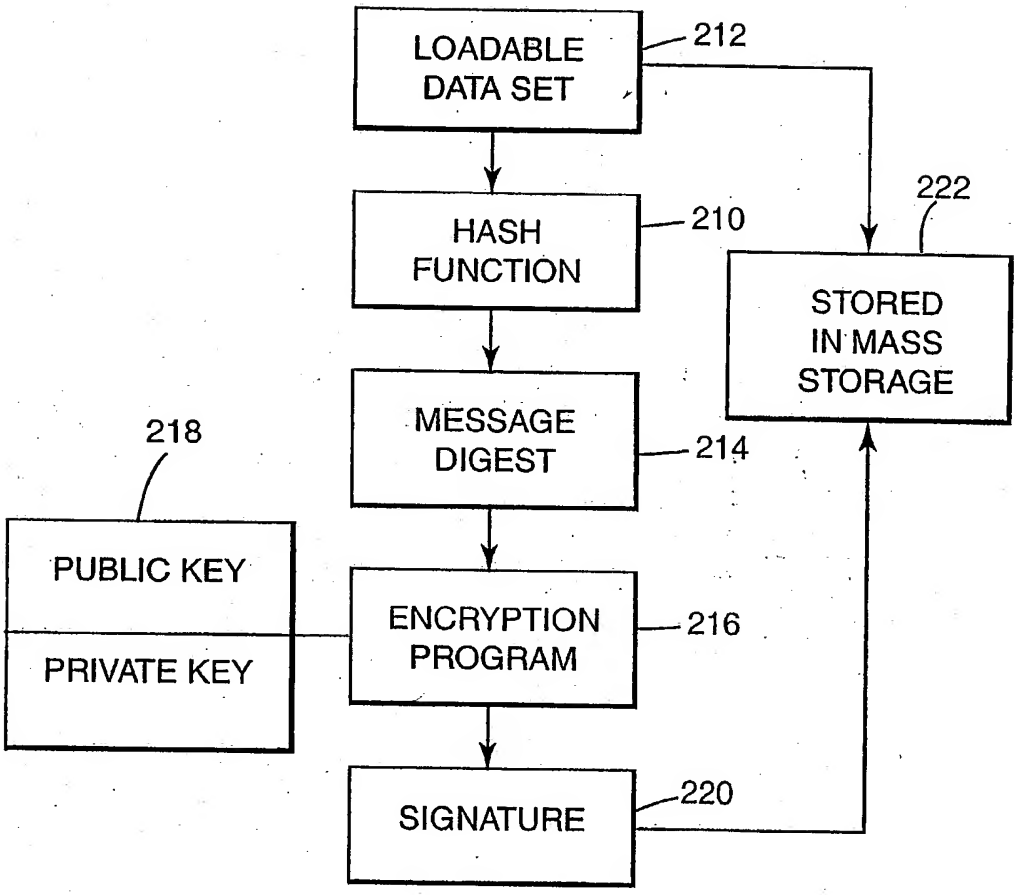
LOADABLE DATA SET
212
HASH FUNCTION
210
MESSAGE DIGEST
214
ENCRYPTION PROGRAM
216
SIGNATURE
220
STORED IN MASS STORAGE
222
218
PUBLIC KEY
PRIVATE KEY

Fig. 3

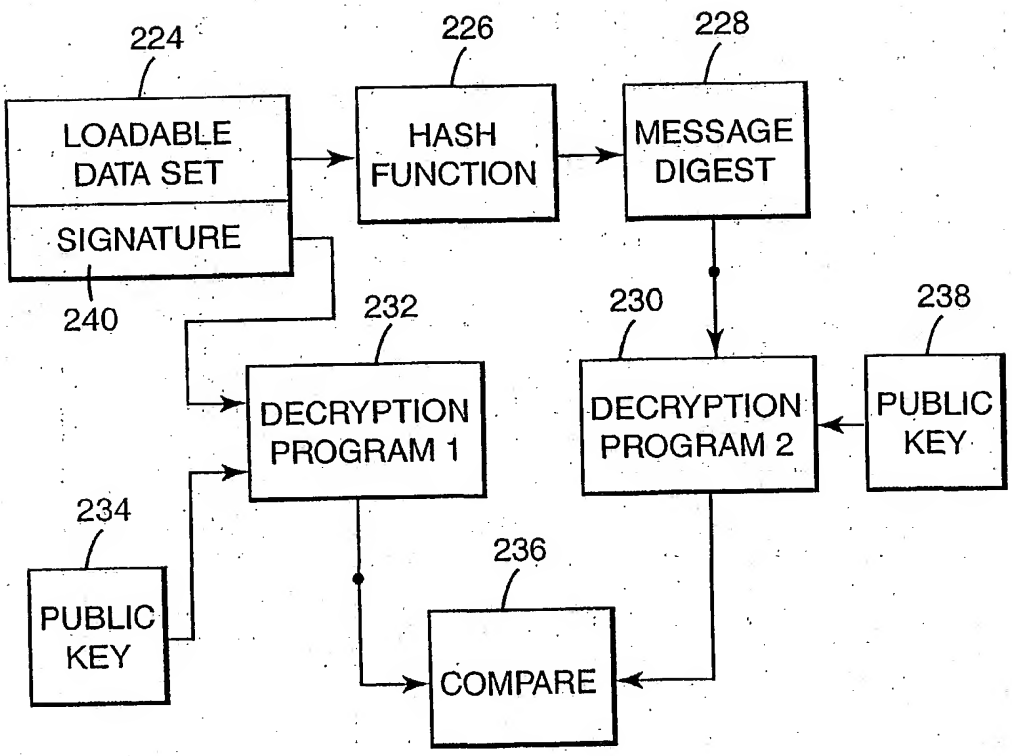
224
LOADABLE DATA SET
SIGNATURE
240
226
HASH FUNCTION
228
MESSAGE DIGEST
232
DECRYPTION PROGRAM 1
230
DECRYPTION PROGRAM 2
238
PUBLIC KEY
234
PUBLIC KEY
236
COMPARE

Fig. 4

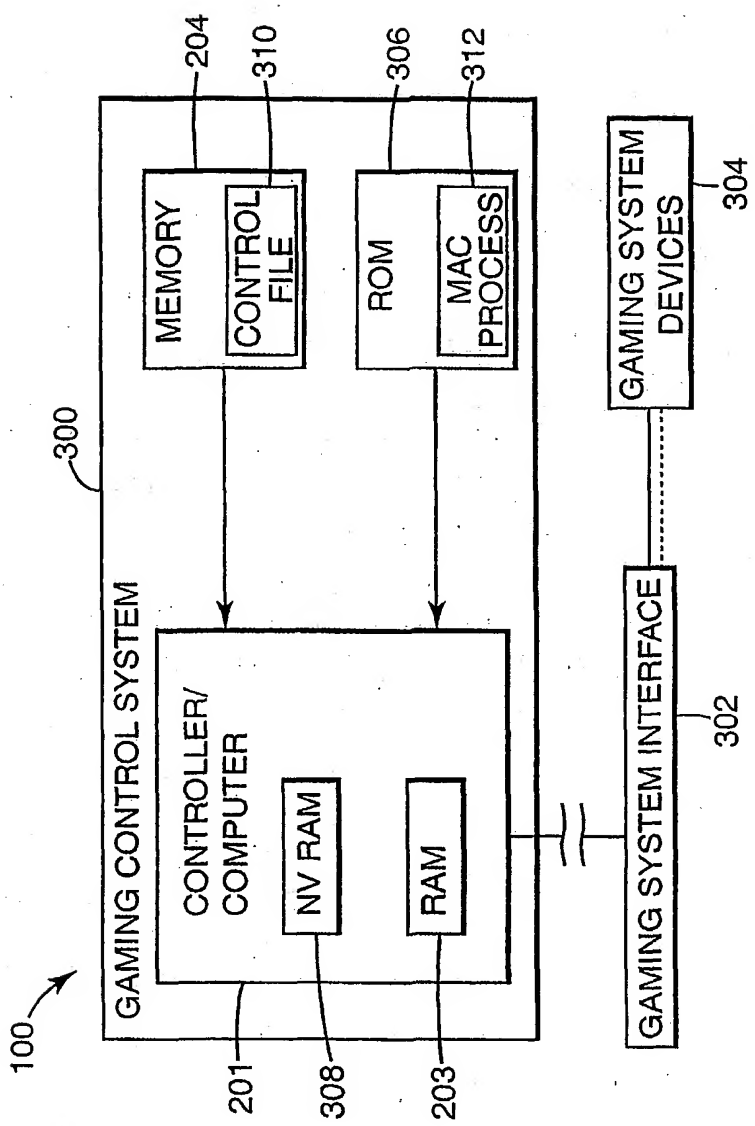
100
GAMING CONTROL SYSTEM
300
CONTROLLER/ COMPUTER
NV RAM
RAM
MEMORY
CONTROL FILE
ROM
MAC PROCESS
204
310
306
312
201
308
203
GAMING SYSTEM INTERFACE
302
GAMING SYSTEM DEVICES
304

Fig. 5

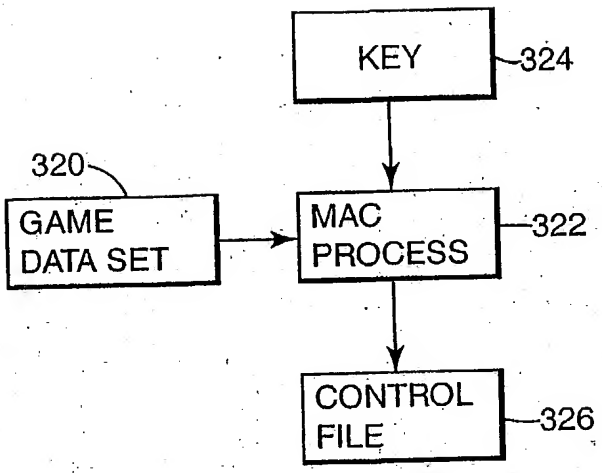
KEY
324
320
GAME DATA SET
MAC PROCESS
322
CONTROL FILE
326

Fig. 6

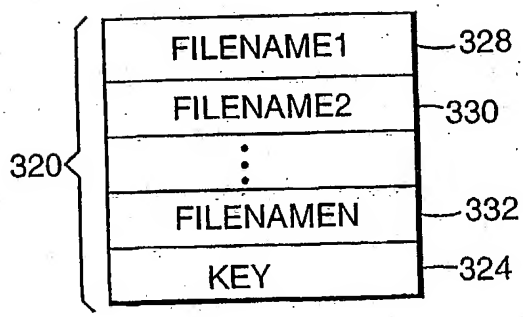
FILENAME1
328
FILENAME2
330
320
FILENAMEN
332
KEY
324

Fig. 7

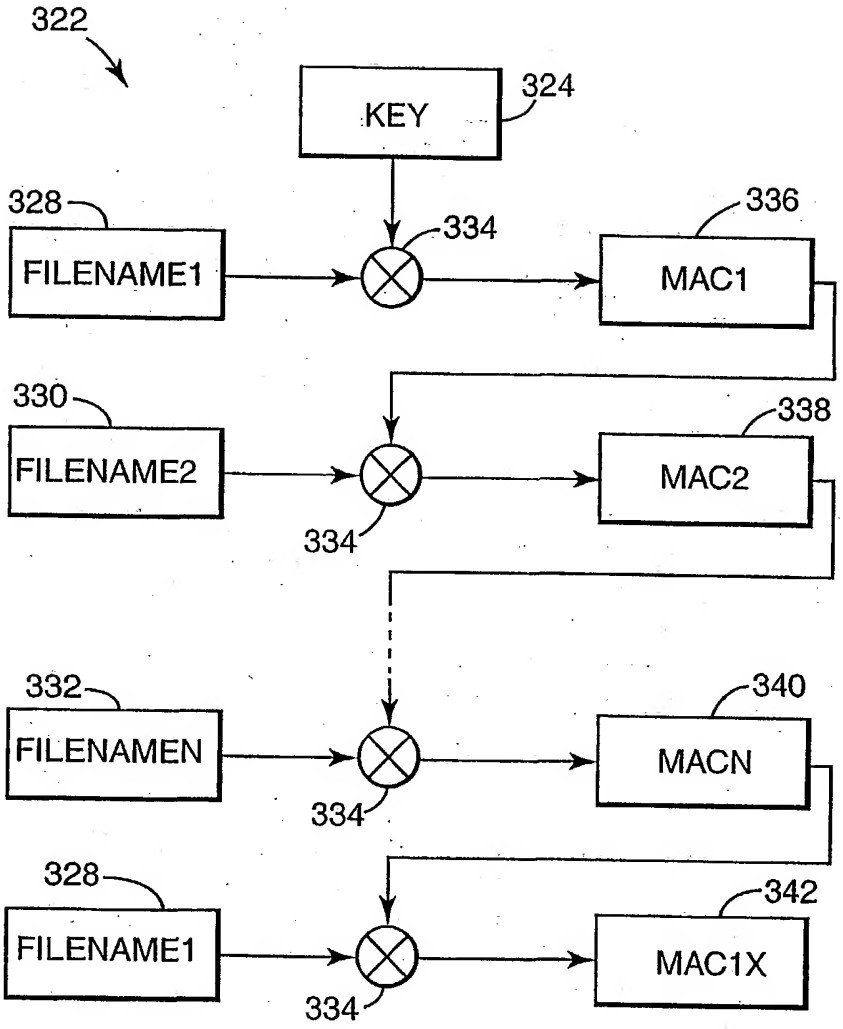
322
KEY
324
328
FILENAME1
334
336
MAC1
330
FILENAME2
334
338
MAC2
332
FILENAMEN
334
340
MACN
328
FILENAME1
334
342
MAC1X

Fig. 8

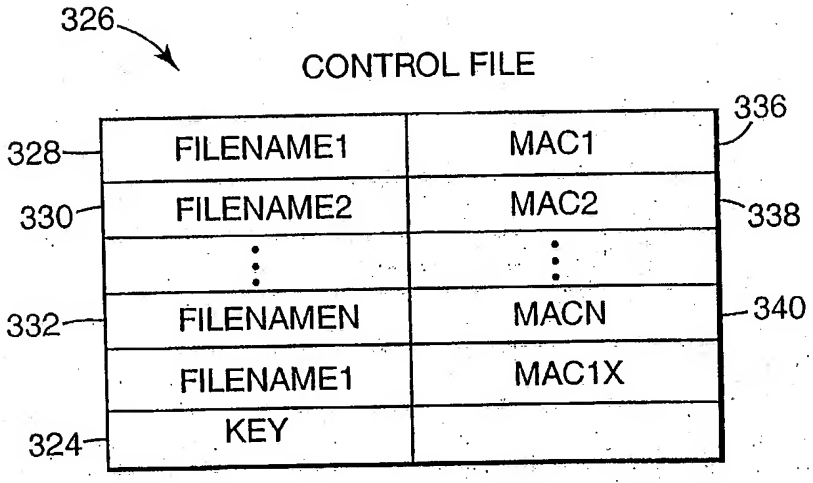
326
CONTROL FILE
336
328
FILENAME1
MAC1
330
FILENAME2
MAC2
338
332
FILENAMEN
MACN
340
FILENAME1
MAC1X
324
KEY

Fig. 9

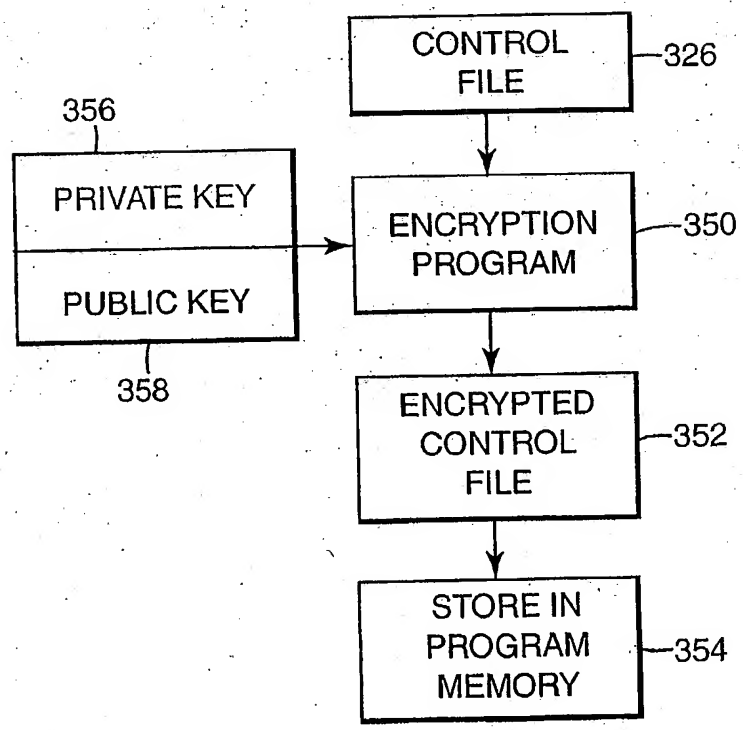
CONTROL FILE
326
356
PRIVATE KEY
PUBLIC KEY
358
ENCRYPTION PROGRAM
350
ENCRYPTED CONTROL FILE
352
STORE IN PROGRAM MEMORY
354

Fig. 10

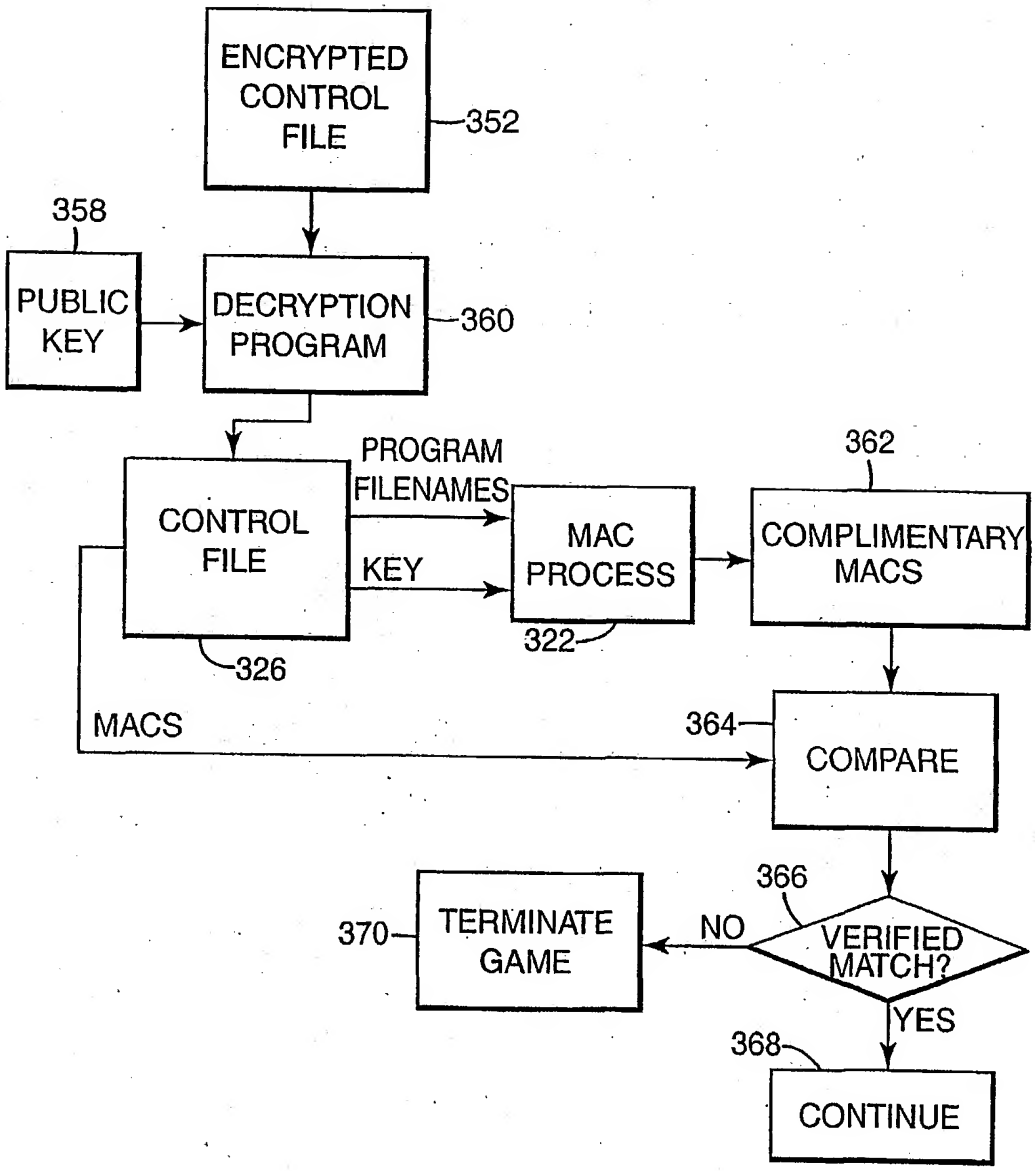
ENCRYPTED CONTROL FILE
352
358
PUBLIC KEY
DECRYPTION PROGRAM
360
PROGRAM FILENAMES
KEY
CONTROL FILE
326
MAC PROCESS
322
362
COMPLIMENTARY MACS
MACS
364
COMPARE
366
VERIFIED MATCH?
NO
370
TERMINATE GAME
YES
368
CONTINUE

Fig. 11

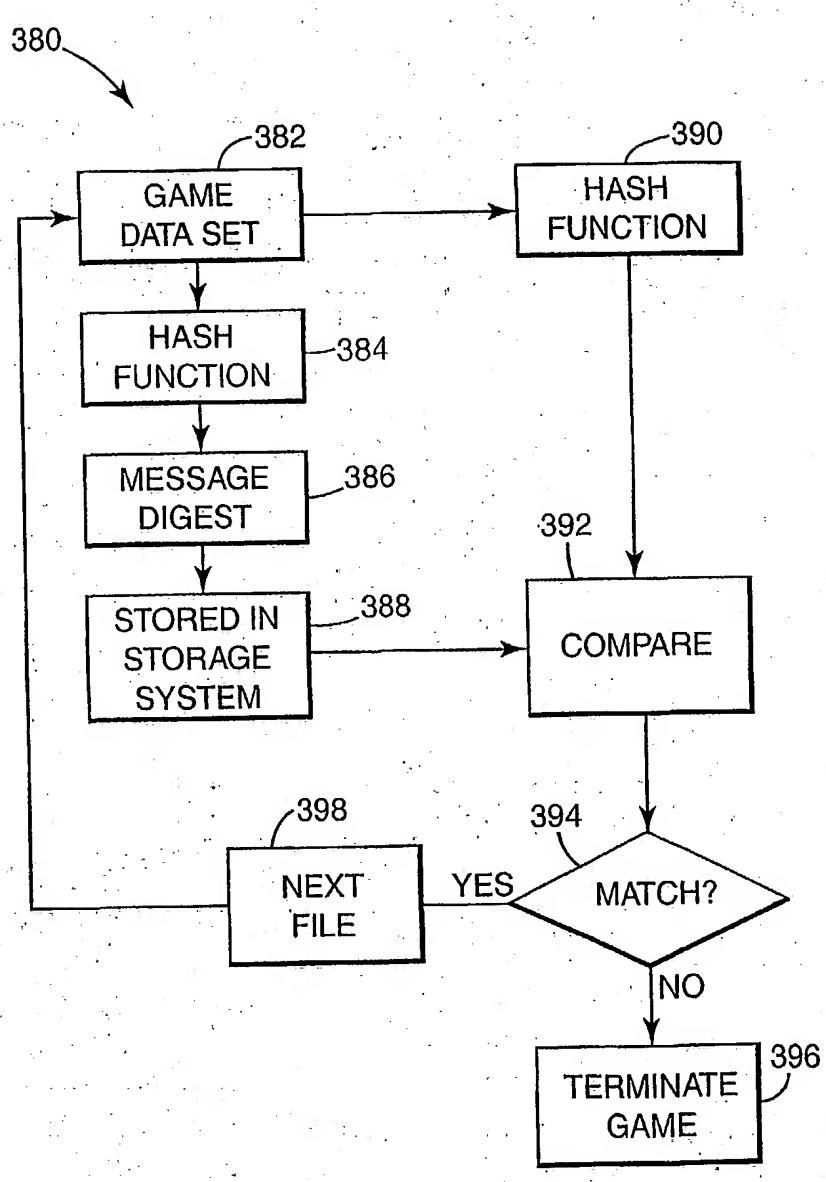
380
382
GAME DATA SET
390
HASH FUNCTION
HASH FUNCTION
384
MESSAGE DIGEST
386
392
STORED IN STORAGE SYSTEM
388
COMPARE
398
394
NEXT FILE
YES
MATCH?
NO
396
TERMINATE GAME

Fig. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/07381

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 1/24
US CL : 713/161, 176, 180
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 713/161, 176, 180

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
West

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,326,104 A (PEASE et al.) 05 July 1994, col. 6, lines 54-66, col. 17, lines 16-23, col. 18, lines 50-65, col. 20, lines 14-40, col. 22, lines 18-32.	1-70

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 May 2001 (01.05.2001)	31 MAY 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Tod Swann Peggy Harrod Telephone No. 703 305-3900

Form PCT/ISA/210 (second sheet) (July 1998)